

AMENDMENTS TO THE CLAIMS

This listing of the claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method for automatically validating text input, the method comprising:

processing a markup language file to receive text input, said markup language file comprising a description of a graphical user interface (GUI), the description comprising a GUI element enabled to receive the text input, wherein the markup language file comprises a markup language tag for instantiating a validation manager;

in response to processing the markup language file, instantiating the validation manager
~~in response to said processing the markup language file;~~

in response to processing the markup language file, displaying the GUI on a display
screen of a computer system ~~in response to said processing the markup language file;~~

receiving text that is input to the GUI element enabled to receive text input;

in response to receiving the text that is input to the GUI element, sending ~~receiving~~ a
programmatic event by to the validation manager ~~in response to said received text input to the~~
~~GUI element;~~

in response to receiving the programmatic event at the validation manager, determining
whether the received text ~~input received to the GUI element~~ is valid text input ~~in response to~~
~~receiving the programmatic event;~~ and

providing an indication that the received text ~~input received to the GUI element~~ is invalid
if the ~~the~~ received text ~~input~~ is determined to be invalid.

2. (Previously Presented) The method of claim 1, wherein the description of the GUI element enabled to receive text input comprises an attribute for specifying a type associated with the GUI element, the method further comprising:

the validation manager instantiating a validation component, wherein said instantiating a validation component comprises specifying the type associated with the GUI element;

wherein said validation manager determining whether the text input at the GUI element is valid text input comprises the validation manager calling the validation component;

wherein said validation manager calling the validation component comprises the validation manager specifying the text input to the GUI element;

wherein the validation component is operable to return a result value to the validation manager indicating whether the text input received to the GUI element is valid text for the type associated with the GUI element.

3. (Previously Presented) The method of claim 1, wherein
the step of receiving text input to the GUI element is performed by a user of the application.

4. (Previously Presented) The method of claim 1,
wherein the step of receiving text input to the GUI element comprises an application providing text input to the GUI element.

5. (Previously Presented) The method of claim 1,
wherein the description of the GUI element enabled to receive text input comprises one or more attributes for specifying the text input to be validated; and
wherein the validation manager is operable to validate text input to the GUI element in accordance with the one or more attributes specifying when text input to the GUI elements should be validated.

6. (Previously Presented) The method of claim 5,
wherein each of the one or more attributes for specifying when text input to the GUI element should be validated corresponds to at least one type of programmatic event; and
wherein the step of said validation manager receiving a programmatic event comprises the validation manager ignoring the programmatic event if the programmatic event does not correspond to one of the attributes for specifying when text input to the GUI element should be validated.

7. (Previously Presented) The method of claim 1,
wherein the step of receiving text input to the GUI element comprises receiving one or more of the following actions:

changing the value of the GUI element,
pressing a key,
releasing a key,
causing the GUI element to gain user interface focus [[.]],
causing the GUI element to lose user interface focus,
moving a mouse pointer over the GUI element,
clicking on the GUI element,
double-clicking on the GUI element;

wherein said validation manager receiving a programmatic event in response to said text input to the GUI element comprises the validation manager receiving a programmatic event corresponding to the action performed.

8. (Previously Presented) The method of claim 1,
wherein the markup language tag for instantiating the validation manager comprises one or more parameters for specifying the default behavior of when the validation manager should validate text input for GUI elements described in the markup language file.

9. (Previously Presented) The method of claim 8,
wherein the description of the GUI element enabled to receive text input comprises one or more attributes for specifying when text input to the GUI element should be validated; and
wherein the validation manager is operable to override the default behavior and validate text input to the GUI element in accordance with the one or more attributes specifying when text input received to the GUI element should be validated.

10. (Previously Presented) The method claim 1,
wherein said validation manager indicating that the text input received to the GUI element is invalid comprises the validation manager requesting the application to alter the visual appearance of the GUI element.

11. (Previously Presented) The method of claim 1,
wherein said validation manager indicating that the text input received to the GUI element is invalid comprises the validation manager displaying an informational user interface window.

12. (Previously Presented) The method of claim 1,
wherein the description of the GUI element enabled to receive text input comprises one or more attributes for controlling text input validation for the GUI element;
wherein said step of processing a markup language file comprises the application constructing a document object representing the markup language file;
wherein instantiating the validation manager comprises the application passing a reference to the document object to the validation manager; and
wherein, in response to being instantiated by and receiving the reference to the documents object, the validation manager is operable to traverse the document object in order to discover the one or more attributes for controlling text input validation for the GUI element.

13. (Original) The method of claim 12,
wherein the one or more attributes for controlling text input validation for the GUI element include an attribute for specifying a type associated with the GUI element.

14. (Original) The method of claim 12,
wherein the one or more attributes for controlling text input validation for the GUI element include one or more an attributes for specifying when text input to the GUI element should be validated.

15. (Original) The method of claim 12,
wherein the one or more attributes for controlling text input validation for the GUI element include one or more attributes for specifying how invalid text input for the GUI element should be indicated.

16. (Previously Presented) The method for claim 1,
wherein the validation manager is a COM object.

17. (Previously Presented) The method of claim 1,
wherein the validation manager is a Java object.

18. (Original) The method of claim 1,
wherein the markup language is HTML.

19. (Previously Presented) The method of claim 2, wherein the type associated with the GUI element is a type comprising one or more of:

Coordination of Benefits (COB) code, Current Procedural Terminology (CPT) code, HCFA Common Procedure Coding System (HCPCS) code, International Classification of Diseases (ICD) code, U.S. Employer Information Number (EIN), U.S. Social Security Number, currency, U.S. states and territories, telephone number, zip code, date, and Boolean.